



Программный комплекс Систэм Платформ

---

SePlatform.Data Server 2.1  
Сервисное приложение  
Просмотрщик лога кадров

---

Руководство администратора

Редакция  
3. Предварительная

Соответствует версии ПО  
2.1.2

---



© ООО «СИСТЭМ СОФТ», 2022-2024. Все права защищены.

Авторские права на данный документ принадлежат ООО «СИСТЭМ СОФТ». Копирование, перепечатка и публикация любой части или всего документа не допускается без письменного разрешения правообладателя.

# Содержание

---

<b>1. Назначение .....</b>	<b>4</b>
<b>2. Просмотр журнала работы модуля .....</b>	<b>5</b>
2.1. Описание свойств записи журнала .....	6
2.2. Описание типов и цветовой индикации записей .....	6
<b>3. Фильтрация и поиск записей .....</b>	<b>8</b>
3.1. Условия типов записей .....	10
3.2. Условия с меткой времени .....	11
3.3. Условия с IP-адресами .....	11
3.4. Условия со строками .....	12
3.5. Условия с числами .....	12
3.6. Сложные логические условия .....	13

# 1. Назначение

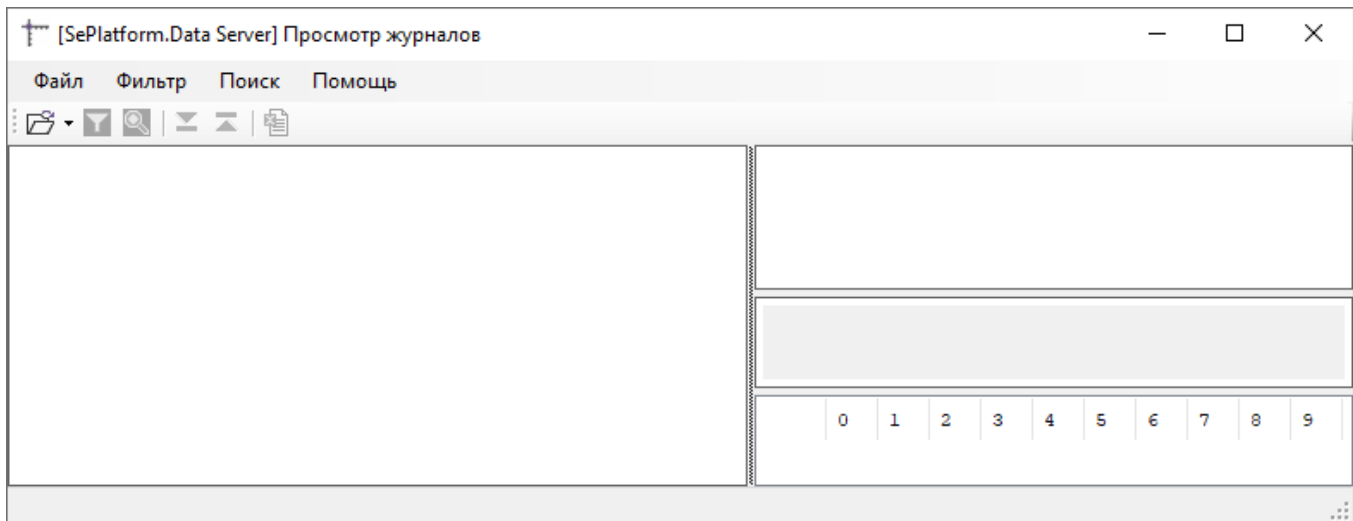
Сервисное приложение Просмотрщик лога кадров предназначено для просмотра журналов работы модулей SePlatform.Data Server.



## ОБРАТИТЕ ВНИМАНИЕ

Сервисное приложение Просмотрщик лога кадров входит в состав клиентской части SePlatform.Data Server и может быть установлено только на компьютере под управлением ОС Windows.

Запустите приложение Просмотрщик лога кадров выполнив команду меню Пуск → SePlatform → Просмотрщик лога кадров, или запустите исполняемый файл SePlatformLogViewer.exe, расположенный в C:\Program Files\SePlatform\SePlatform.Server\Service.



Основными функциями приложения являются:

- отображение журнала работы модуля с обновлением данных в режиме реального времени;
- фильтрация записей;
- поиск записей;
- сохранение сообщений о работе модуля в табличный файл.

В журнале работы модуля отображаются:

- сообщения о работе модуля;
- технологические данные, с которыми работает модуль.


Приложение позволяет экспортировать журналы работы модуля в файл с расширением \*.csv. Для экспорта журнала:

1. Откройте журнал работы модуля.
2. На панели инструментов нажмите кнопку
3. В окне **Настройка экспорта** выберите столбцы, которые будут отображены в файле, и нажмите **Экспорт**.
4. В окне **Сохранить** как введите имя файлу и нажмите **Сохранить**.


## 2. Просмотр журнала работы модуля

Ведение журнала работы модуля настраивается в общих параметрах модуля.

Для открытия журнала работы модуля:

1. В меню **Файл** выберите **Открыть**, либо нажмите кнопку  или сочетание клавиш «Ctrl»+«O».
2. В появившемся окне **Открыть** выберите журнал работы модуля и нажмите **Открыть**. Файлы журналов модулей имеют расширение \*.aplog и находятся по умолчанию:
  - в ОС Windows в папке C:\Program Files\SePlatform\SePlatform.Server\Logs;
  - в ОС Linux в директории /opt/SePlatform/Logs.

Расположение журналов модулей указывается в файле SePlatform.Server.xml в атрибуте **Path** элемента **Log**. Файл SePlatform.Server.xml расположен в папке/директории установки SePlatform.Data Server.

Чтобы выбрать файл журнала из списка последних просмотренных файлов, нажмите на стрелочку справа от кнопки .

В окне приложения отобразятся записи журнала работы модуля в виде таблицы со столбцами, которые соответствуют набору свойств модуля.



Файл    Фильтр    Поиск    Помощь			
№	Дата	Время	Описание
0	18.04.2021	10:18:54:750	Журнал открыт для записи
1	18.04.2021	10:18:54:750	--> CModule::OnInitialize
2	18.04.2021	10:18:54:820	<-- CModule::OnInitialize
3	18.04.2021	10:18:55:300	--> CModule::OnWorkingMode
4	18.04.2021	10:18:55:300	<-- CModule::OnWorkingMode
5	18.04.2021	10:18:55:410	--> CModule::OnStart
6	18.04.2021	10:18:55:430	<-- CModule::OnStart
7	18.04.2021	10:20:12:734	--> CModule::OnStop
8	18.04.2021	10:20:12:744	<-- CModule::OnStop
9	18.04.2021	10:20:12:824	--> CModule::OnUninitialize
10	18.04.2021	10:20:12:824	<-- CModule::OnUninitialize
11	18.04.2021	10:20:12:824	Журнал закрыт
12	18.04.2021	10:20:15:311	Журнал открыт для записи

Журнал открыт для записи

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

Количество записей: 387

Для переходов между сессиями журнала модуля используйте кнопки:

-  - переход к началу следующей сессии;
-  - переход к началу текущей сессии.

Области в правой части окна приложения предназначены для отображения кадров данных, с которыми работает модуль:

- расшифровка полей кадра данных;
- описание выбранной записи;

- побайтовое представление кадра данных в шестнадцатеричном виде.

расшифровка полей кадра данных →

№	Адрес	Значение	Метка времени	Качество	Номер бита	Тип
1	0	22.03.2021 5:23:01				Синхронизация

описание выбранной записи →

Исходящий кадр <103> Синхронизация (S:0;R:0)

побайтовое представление кадра данных →

	0	1	2	3	4	5	6	7	8	9
0000	68	14	00	00	00	00	67	01	06	00
0001	01	00	00	00	00	56	06	17	05	96
0002	03	0C								

Расшифровка полей кадра данных отображается в виде таблицы со столбцами, соответствующими полям кадра. Набор полей кадра данных зависит от конкретного типа модуля.

## 2.1. Описание свойств записи журнала

Каждая запись в журнале модуля описывается набором свойств. По значению свойств записей производится фильтрация и поиск записей в журнале модуля.

Для журналов модулей существует набор свойств:

- **Description** - описание записи. Тип: строка (System.String), поле таблицы: **Описание**;
- **Type** - полный тип записи. Тип: число (System.UInt16);
- **StdType** - стандартизированный тип записи. Является подмножеством Type. Тип: перечисление Axidep.LogViewer.LogRecordType;
- **TimeStamp** - метка времени, когда была произведена запись. Тип: дата и время (System.DateTime), поля таблицы: **Дата и Время**;
- **RecordNum** - номер записи в журнале. Тип: число (System.Int32), поле таблицы: **№**.

Набор свойств, отображаемых в журнале модуля, зависит от модуля.

## 2.2. Описание типов и цветовой индикации записей

Записи журнала, отображаемые в окне приложения, выделяются различными цветами в зависимости от типа записи. Тип записи определяется свойством **StdType**. Стандартные типы записей и цветовая индикация записей:

- **Error** - ошибка (темно-красный);
- **Warning** - предупреждение (темно-оранжевый);
- **Info** - информация (синий);
- **Debug** - отладка (темно-серый);
- **Start** - журнал открыт для записи (зеленый);
- **Stop** - журнал закрыт (оранжевый);
- **Enable** - запись в журнал разрешена (черный);
- **Disable** - запись в журнал запрещена (черный);
- **IncomingChannelFrame** - входящий кадр (темно-зеленый);
- **OutgoingChannelFrame** - исходящий кадр (темно-синий);



- **Connected** - соединение установлено (серый);
- **Disconnected** - соединение разорвано (серый);
- **ClientConnected** - подключился клиент (серый);
- **ClientDisconnected** - отключился клиент (серый);
- **FunctionNameEnter** - вход в функцию (серый);
- **FunctionNameExit** - выход из функции (серый).

## 3. Фильтрация и поиск записей

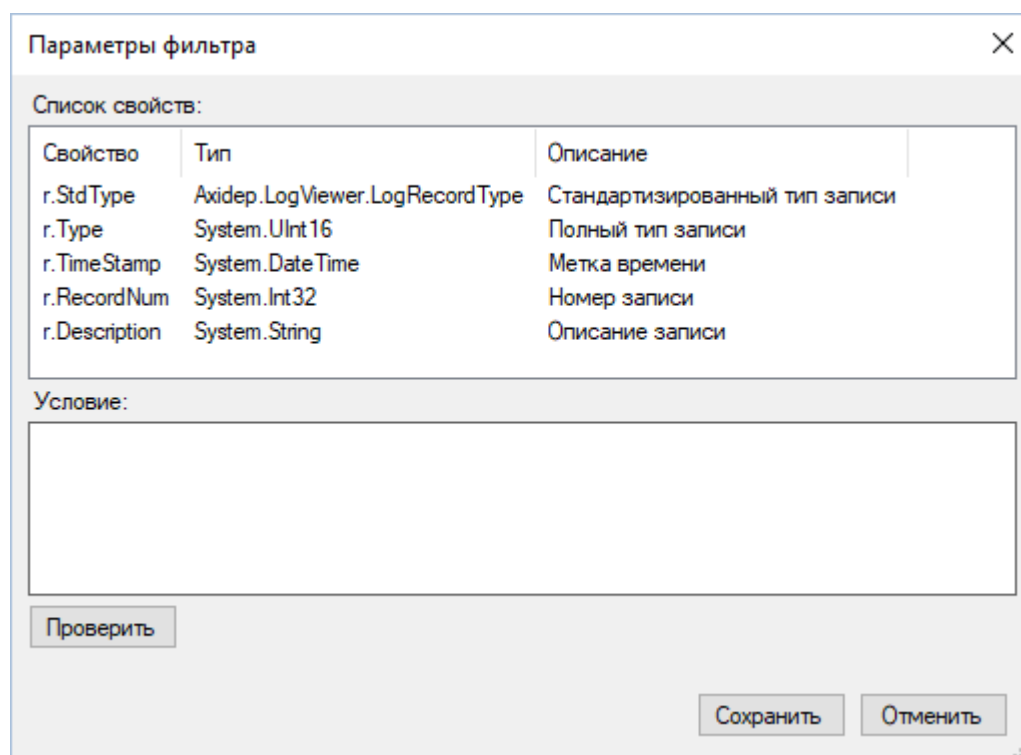
Для удобства анализа работы модуля и поиска записей используйте фильтрацию и поиск записей.

Выражения условий записываются с использованием синтаксиса языка C#.

Для установки параметров фильтрации записей или поиска записей:

- в меню **Фильтр** выберите **Параметры фильтра**, либо нажмите кнопку  или сочетание клавиш «Alt»+«F».
- в меню **Поиск** выберите **Найти**, либо нажмите кнопку  или сочетание клавиш «Ctrl»+«F».

В результате отобразится окно **Параметры фильтра** или **Параметры поиска**.



Параметры фильтра

Список свойств:

Свойство	Тип	Описание
r.StdType	Axidep.LogViewer.LogRecordType	Стандартизированный тип записи
r.Type	System.UInt16	Полный тип записи
r.TimeStamp	System.DateTime	Метка времени
r.RecordNum	System.Int32	Номер записи
r.Description	System.String	Описание записи

Условие:

Проверить

Сохранить Отменить



Параметры поиска

Список свойств:

Свойство	Тип	Описание
r.StdType	Axidep.LogViewer.LogRecordType	Стандартизированный тип записи
r.Type	System.UInt16	Полный тип записи
r.TimeStamp	System.DateTime	Метка времени
r.RecordNum	System.Int32	Номер записи
r.Description	System.String	Описание записи

Условие:

Проверить

Сохранить Отменить

В окнах отображаются поля **Список свойств:** и **Условие:**, предназначенные для указания параметров фильтрации или поиска записей в журнале модуля.

В поле **Список свойств:** отображается перечень доступных свойств записей текущего журнала модуля.

Поле **Условие:** предназначено для ввода выражения условий фильтра или поиска. В журнале отображаются записи, для которых выражение фильтра или поиска принимает значение «true». Пустая строка условия означает отсутствие фильтрации или поиска.

Чтобы добавить свойства записи в поле **Условие:**, в таблице **Список свойств:** дважды кликните мышью по свойству.

Правила создания условий для фильтрации или поиска записей:

- условие задается в виде:

<имя свойства> <оператор сравнения> <значение>

- для создания условия используйте операторы сравнения: ==, >, <, !=, >=, <=;
- выражение условия может состоять из одного или нескольких условий, соединенных с помощью логических связок И, ИЛИ, НЕ. Для создания логических связок используйте операторы: && - и, || - или, ! - не;
- для указания приоритетов операций используйте скобки;
- строковые значения заключайте в двойные кавычки;
- имена операторов чувствительны к регистру написания.



#### ОБРАТИТЕ ВНИМАНИЕ

Значение «null» могут принимать все свойства, тип которых не является числом или перечислением. Если тип свойства в поле **Список свойств:** помечен знаком «?», то данное свойство может принимать значение «null».

Чтобы проверить правильность логического выражения, введенного в поле **Условие:**, в окне **Параметры фильтра** или **Параметры поиска** нажмите **Проверить**. В результате отобразится надпись «Условие корректно» или «Условие не корректно».

## 3.1. Условия типов записей

Свойство `StdType` - стандартизированный тип записи, относится к типу `Axidep.LogViewer.LogRecordType`.

Примеры:

1. Вывести все записи с типом ошибка:

```
r.StdType == Axidep.LogViewer.LogRecordType.Error
```

2. Вывести все записи с типом предупреждение:

```
r.StdType == Axidep.LogViewer.LogRecordType.Warning
```

3. Вывести все записи с типом информация:

```
r.StdType == Axidep.LogViewer.LogRecordType.Info
```

4. Вывести все записи с типом отладка:

```
r.StdType == Axidep.LogViewer.LogRecordType.Debug
```

5. Вывести все записи с типом журнал открыт для записи:

```
r.StdType == Axidep.LogViewer.LogRecordType.Start
```

6. Вывести все записи с типом журнал закрыт:

```
r.StdType == Axidep.LogViewer.LogRecordType.Stop
```

7. Вывести все записи с типом запись в журнал разрешена:

```
r.StdType == Axidep.LogViewer.LogRecordType.Enable
```

8. Вывести все записи с типом запись в журнал запрещена:

```
r.StdType == Axidep.LogViewer.LogRecordType.Disable
```

9. Вывести все записи с типом входящий кадр:

```
r.StdType == Axidep.LogViewer.LogRecordType.IncomingChannelFrame
```

10. Вывести все записи с типом исходящий кадр:

```
r.StdType == Axidep.LogViewer.LogRecordType.OutgoingChannelFrame
```

11. Вывести все записи с типом соединение установлено:

```
r.StdType == Axidep.LogViewer.LogRecordType.Connected
```

12. Вывести все записи с типом соединение разорвано:

```
r.StdType == Axidep.LogViewer.LogRecordType.Disconnected
```

13. Вывести все записи с типом подключился клиент:

```
r.StdType == Axidep.LogViewer.LogRecordType.ClientConnected
```

14. Вывести все записи с типом отключился клиент:

```
r.StdType == Axidep.LogViewer.LogRecordType.ClientDisconnected
```

15. Вывести все записи с типом вход в функцию:

```
r.StdType == Axidep.LogViewer.LogRecordType.FunctionNameEnter
```

16. Вывести все записи с типом выход из функции:

```
r.StdType == Axidep.LogViewer.LogRecordType.FunctionNameExit
```

## 3.2. Условия с меткой времени

Свойство **TimeStamp** - метка времени записи относится к типу **System.DateTime**.

Условие задается в виде:

```
r.TimeStamp <знак сравнения> new System.DateTime (Yyyy, Mm, Dd, hh, mm, ss);
```

где **Yyyy** - год, **Mm** - месяц, **Dd** - день, **hh** - часы, **mm** - минуты, **ss** - секунды.

Примеры:

1. Отобразить все записи за сегодняшний день:

```
r.TimeStamp >= System.DateTime.Today
```

2. Отобразить записи, которые были добавлены после 10.05.2020:

```
r.TimeStamp > new System.DateTime (2020, 05, 10)
```

3. Отобразить записи, которые были добавлены в период с 10.05.20 14:20:00 по 11.05.20 7:00:00:

```
r.TimeStamp >= new System.DateTime(2020, 05, 10, 14, 20, 00 ) && r.TimeStamp <= new System.DateTime(2020, 05, 11, 7, 00, 00)
```

## 3.3. Условия с IP-адресами

IP-адреса являются значениями типа **System.Net.IPAddress**.

Пример: Отобразить все записи, для которых задан IP-адрес «127.0.0.1»:

```
r.RemoteIp != null && r.RemoteIp.Address == System.Net.IPAddress.Parse("127.0.0.1").Address
```

## 3.4. Условия со строками

Для создания условий с использованием строковых значений (тип `System.String`) предназначены функции:

- **Contains** - значение свойства содержит указанную подстроку. Условие задается в виде:

```
r.Description.Contains("<подстрока>")
```

- **StartsWith** - значение свойства начинается с указанной подстроки. Условие задается в виде:

```
r.Description.StartsWith("<подстрока>")
```

- **EndsWith** - значение свойства заканчивается указанной подстрокой. Условие задается в виде:

```
r.Description.EndsWith("<подстрока>")
```

Примеры:

1. Отобразить записи, описание которых равно «Входящий кадр Транзакция 54943, Функция 1, Ошибка код 2»:

```
r.Description=="Входящий кадр Транзакция 54943, Функция 1, Ошибка код 2"
```

2. Отобразить записи, описание которых содержит подстроку «Функция 3»:

```
r.Description.Contains("Функция 3")
```

3. Отобразить записи, которые начинаются с подстроки «Исходящий кадр»:

```
r.Description. StartsWith("Исходящий кадр")
```

4. Отобразить записи, описание которых не зависит от регистра и начинается с подстроки «Количество»:

```
r.Description.StartsWith("количество", StringComparison.OrdinalIgnoreCase)
```

5. Отобразить записи, которые заканчиваются подстрокой «код 2»:

```
r.Description.EndsWith ("код 2")
```



### ОБРАТИТЕ ВНИМАНИЕ

При указании адреса ставьте пробел после числа.

Более подробно о функциях типа строка <http://msdn.microsoft.com/ru-ru/library/system.string.aspx>.

## 3.5. Условия с числами

Работа с типами `System.Byte`, `System.Int16`, `System.UInt16`, `System.Int32` является общепринятой работой с численными типами. Используемые операторы для фильтрации: `>`, `<`, `==`, `>=`, `<=`.

Пример: Отобразить только записи, относящиеся к станции 1.

```
r.Station==1
```

## 3.6. Сложные логические условия

Примеры:

1. Отобразить записи входящих кадров, начиная с даты «10.05.2020» с «16:00:00», выполняющих функцию «3» и полученных от станции с IP-адресом «127.0.0.1»:

```
(r.Description.Contains("Функция 3")) && (r.Description.Contains("Входящий кадр")) &&  
(r.TimeStamp > new System.DateTime(2020, 05, 10, 16, 0, 0)) && (r.RemoteIp.Address ==  
System.Net.IPAddress.Parse("127.0.0.1").Address)
```

2. Отобразить записи с типом ошибка, исключая дату «21 апреля 2020 года», и записи с типом предупреждение, начиная с даты «20 апреля 2020 года»:

```
((r.TimeStamp != new System.DateTime(2020, 04, 21)) && (r.StdType ==  
Axidep.LogViewer.LogRecordType.Error)) || ((r.TimeStamp >= new System.DateTime(2020, 04,  
20)) && (r.StdType == Axidep.LogViewer.LogRecordType.Warning))
```

3. Отобразить записи до «19 апреля 2020 года» с типами вход в функцию и выход из функции, сгруппированные по сессиям.

```
(r.TimeStamp <= new System.DateTime(2020, 04, 19)) && ((r.StdType ==  
Axidep.LogViewer.LogRecordType.FunctionNameEnter) || (r.StdType ==  
Axidep.LogViewer.LogRecordType.FunctionNameExit) || (r.StdType ==  
Axidep.LogViewer.LogRecordType.Start) || (r.StdType ==  
Axidep.LogViewer.LogRecordType.Stop))
```